**Game Lords**

## Description:

* The App builds for the playing html5 games which user having the access code of this application.
* Application contains the great designing and graphics. A user can rename his/her personal information and upload the profile picture.
* A user can play the html5 game on his/her android or ios devices.
* A user can give as favorite and rate particular the html5 game.
* A user can search the game based on the game title name.
* A user can view the game category wise such as Action, Adventure, Premium games, Arcade, and Sports etc.
* A user can view a full description of the html5 games.
* Node js use as a backend for maintaining the data into a database of the application
* To access gamelords application uid is mandatory. Each uid is considered as different user and while accessing site this uid is used to store user related information like user’s favourite games, games liked by user.

**Modules**

Application is divided into number of modules. These modules helps user to find games easily.

1. **HTML5 Games:**

* All HTML5 games are assigned to particular category. Under this section games are listed by category. On game detail page user can view game detail, play game online and view related games.

2. **Favourite**

User can mark games as favourite and Under favourite section he can view all the games marked as favourite.

3. **My Account**

A user can rename his/her personal information and upload the profile picture.

## Mobile Technology:

We are using React Native as the platform to develop the mobile apps for Android, and iOS.

React Native lets you build mobile apps using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components.

**Documentation:** [https://facebook.github.io/react-native/docs/0.55/getting-started.html](https://facebook.github.io/react-native/docs/0.40/getting-started.html)

**Version:** 0.55.4

**Folder Structure:**

|  |  |
| --- | --- |
| **android** | Native Android project files |
| **app** | JS files used to create the app logic |
| **ios** | Native iOS project files |

**App Folder Structure:**

|  |  |
| --- | --- |
| **actions** | Actions which defines an object that describes what happened. |
| **assets** | Fonts and Images used in the app |
| **components** | Contains small components that are used in different screens |
| **constants** | Global Size, Base API, Messages |
| **containers** | Screens files |
| **reducers** | Determine how the state will change. |
| **Store** | Shared store configurations |
| **style** | App styles |
| **util** | Other utilities libraries |

**Data Management:**

We use Redux for global app data management.

Redux is one of the hottest libraries for front-end development in today’s marketplace. It is a predictable state container for JavaScript applications and is used for the entire applications state management. Applications developed with Redux are easy to test and can run in different environments showing consistent behavior.

For more info [Click Here](https://redux.js.org/)

**Git:**

**​**Git is used to manage source code versioning.

**Git repo:**

**https://git.sam-media.com/content/gamelord-reactnative.git**

**API List:**

**BASE API URL:** <http://play.game-lords.com>

**Test Token Code:** fdf098fcc6

|  |  |
| --- | --- |
| **Login** | /checkAccess?uid=fdf098fcc6 |
| **Get User Profile** | /getUserProfile |
| **Get all html5 Games** | /getGames?categoryTypeId=1 |
| **Get Game Details** | /getGames?gameId=`game\_id` |
| **Get Categories** | /getCategories?categoryTypeId=1 |
| **Get interests** | /interests |
| **Get all Favorites of authenticate User** | /getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |
| **Get User Likes Game** | /getUserLikes?uid=`user\_id` |
| **Set Game As Like** | /userLike?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |
| **Upload Profile Pic** | /uploadProfilePic?image=`data:image/jpeg;base64` |
| **Edit Profile** | /editProfile?name=`name`&email=`email`&address=`address`&interests=`interests` |



**Third Party Packages:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Description** |
| **axios** | **0.18.0** | Promise based HTTP client for the browser |
| **react-native-billing** | **2.7.0** | Provide an easy interface to InApp Billing on Android |
| **react-native-device-info** | **0.21.5** | Device Information. |
| **native-base** | **2.3.9** | Essential cross-platform UI components. |
| **react-native-image-picker** | **0.26.7** | Allows you to use native UI to select a photo/video from the device library or directly from the camera. |
| **react-native-in-app-utils** | **6.0.0** | A react-native wrapper for handling in-app payments. |
| **react-native-message-bar** | **2.4.0** | Gradient color. |
| **react-native-orientation** | **3.1.3** | Listen to device orientation changes. |
| **react-native-splash-screen** | **3.1.0** | A splash screen for iOS and Android. |
| **react-native-vector-icons** | **4.5.0** | Customizable Icons. |
| **react-navigation** | **1.4.0** | Routing and navigation. |
| **react-redux** | **5.0.7** | Bindings for Redux. |
| **redux** | **3.7.2** | Predictable state container for JavaScript app. |
| **redux-promise** | **0.5.3** | Promise middleware for Redux. |

**JS File:**

1. **App:**

App is called in index.js to register our application.

We had bind Root Navigator with store to use Redux in our application.

It also contain Navigator Screens and their respective call

1. **SplashScreen:**

This screen is to show the splash image in background till the application get load.

In background a check is done to see the user is logged in or not.

1. **Login:**

For the new user he will be redirected to this screen.

User is asked to add access token for login.

User access-token is stored in Asyncstorage for local storage.

For new user subscription option is provided at the bottom where he can subscribe.

For android - Google play in-app purchase is used. (plugin used: react-native-billing ).

For ios - Apple in-app purchase is used. (InAppUtils from react-native Native Modules ).

API use as follow in this screen,

|  |  |
| --- | --- |
| **Login** | /checkAccess?uid=fdf098fcc6 |

1. **Home:**

User can view Banner, Category Slider and List for Games slider based on category wise.

User can set mark as favorite, and view all favorite game list as well. A user can view all category’s games list.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get User Profile** | /getUserProfile |
| **Get all html5 Games** | /getGames?categoryTypeId=1 |
| **Get Game Details** | /getGames?gameId=`game\_id` |
| **Get Categories** | /getCategories?categoryTypeId=1 |
| **Get interests** | /interests |
| **Get all Favorites of authenticate User** | /getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Favorite:**

User can view his/her own favorite game list. Also, User can set mark as favorite.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get all Favorites of authenticate User** | /getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Category**

User can view game list which category selected.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get all Favorites of authenticate User** | /getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Search:**

It display autosuggest user and allow user to search games.

Search is done locally without api call.On click of search result search text is pushed into *this.props.searchText()* that is used to show data in search screen. User can also set the Set Game as favorite.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Set Game As Favorite** | /favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **Detail:**

User can get all the information regarding the game. Also, User can play the game on his/her device. User can view the related the game list slider which belongs category

API use as follow in this screen,

|  |  |
| --- | --- |
| **Get all html5 Games** | /getGames?categoryTypeId=1 |
| **Get Game Details** | /getGames?gameId=`game\_id` |
| **Get all Favorites of authenticate User** | /getFavorites?uid=`user\_id` |
| **Set Game As Favorite** | /favorite?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |
| **Get User Likes Game** | /getUserLikes?uid=`user\_id` |
| **Set Game As Like** | /userLike?uid=`user\_id`&gameId=`game\_id`&isFavorite=`boolean` |

1. **My Account:**

**componentWillMount:** In this API call is done to get user detail using api *getUserProfile.* userDetail is stored in state.

**\_OpenGallery**: This function contain the call to open gallery. User can update profile pic on click of edit profile. *React-native-image-picker* library is used to open gallery & camera from where user can update profile pic.

**submitForm:** function is used to submit user data at the sever end. *editProfile* api is used to edit user information.

API use as follow in this screen,

|  |  |
| --- | --- |
| **Upload Profile Pic** | /uploadProfilePic?image=`data:image/jpeg;base64` |
| **Edit Profile** | /editProfile?name=`name`&email=`email`&address=`address`&interests=`interests` |

**Redux**:

Redux implementation is done in app. It has all actions inside action folder which also had action type file. Reducer functions are included inside reducer folder. All reducer are combine inside index.js file inside reducer folder.

**Ios certificate:** ios certificate, provisioning file are inside certificate folder.

**Android keystore:** android keystore for generating signed apk is at following path: *‘GameLords/android/app/mobiotv.keystore’*